

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

MASTEROBJECTS, INC.,

Plaintiff,

v.

META PLATFORMS, INC.,

Defendant.

No. C 21-05428 WHA

ORDER RE SUMMARY JUDGMENT

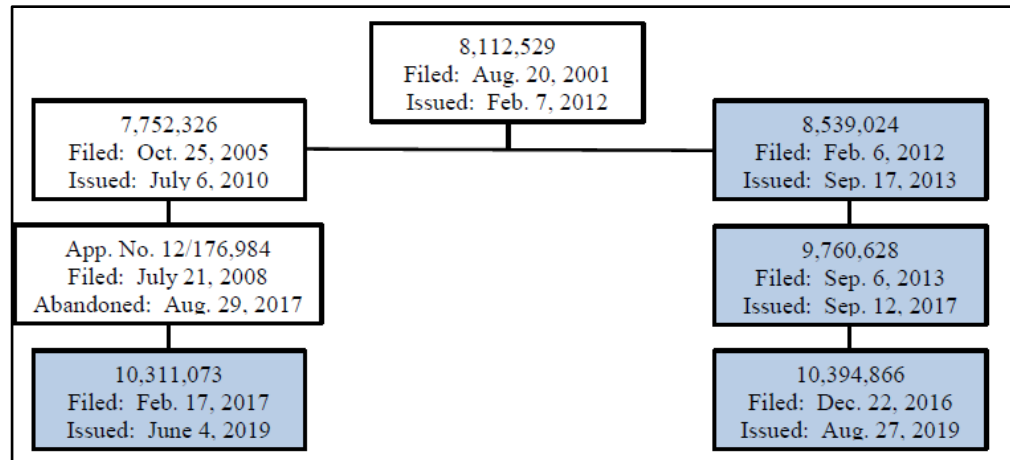
INTRODUCTION

In this patent-infringement action, defendant moves for summary judgment of noninfringement and Section 101 invalidity. To the following extent, because the accused system does not infringe the claims-in-suit as properly construed, the motion is **GRANTED** as to noninfringement. This order need not reach the validity question.

STATEMENT

This litigation concerns autocompletable technology for digital searches. Autocompletion suggests ways for the user to complete her search as she actively types it into a search bar. Patent owner MasterObjects, Inc. accuses alleged infringer Meta Platforms, Inc. (formerly Facebook) of infringing 43 claims across four patents: U.S. Patent Nos. 8,539,024; 9,760,628; 10,311,073; and 10,394,866.

The four patents-in-suit all descend from U.S. Patent No. 8,112,529 (filed in 2001). The following diagram lays out the patent genealogy. The asserted patents are highlighted in blue:



The '024, '628, and '866 patents are sequential continuations of the original '529 patent and share its specification. The '073 patent is a continuation-in-part of the '529 patent and incorporates its specification by reference. The common specification describes an embodiment of the invention, “QuestObjects,” which MasterObjects now labels as “baroquely detailed” and non-limiting (MasterObjects Claim Const. Br. 2).

MasterObjects originally brought this action in the United States District Court for the Western District of Texas, Waco Division, in February 2020. The parties litigated in Texas for seventeen months prior to transfer to our district. During that time, Judge Alan Albright held a *Markman* hearing and, in a minute order, kept his preliminary claim constructions that the parties had debated during oral argument (Dkt. No. 69; Homrig Decl. Exh. 2). Judge Albright construed two terms, “asynchronous” and “query message.”

To frame the analysis that follows, here is claim 1 of the '024 patent, which exemplifies how both of those terms are used in the claims-in-suit (emphasis added):

1[pre] A system comprising:

1[a] a server system, including one or more computers, which is configured to receive query messages from a client object, *the server system asynchronously receiving and responding to the query messages from the client object over a network;*

1[b] the client object that, while a user is providing input comprising a lengthening string of characters, sends query

messages to the server system;

1[c] *whereby the query messages represent the lengthening string as additional characters are being input by the user; and*

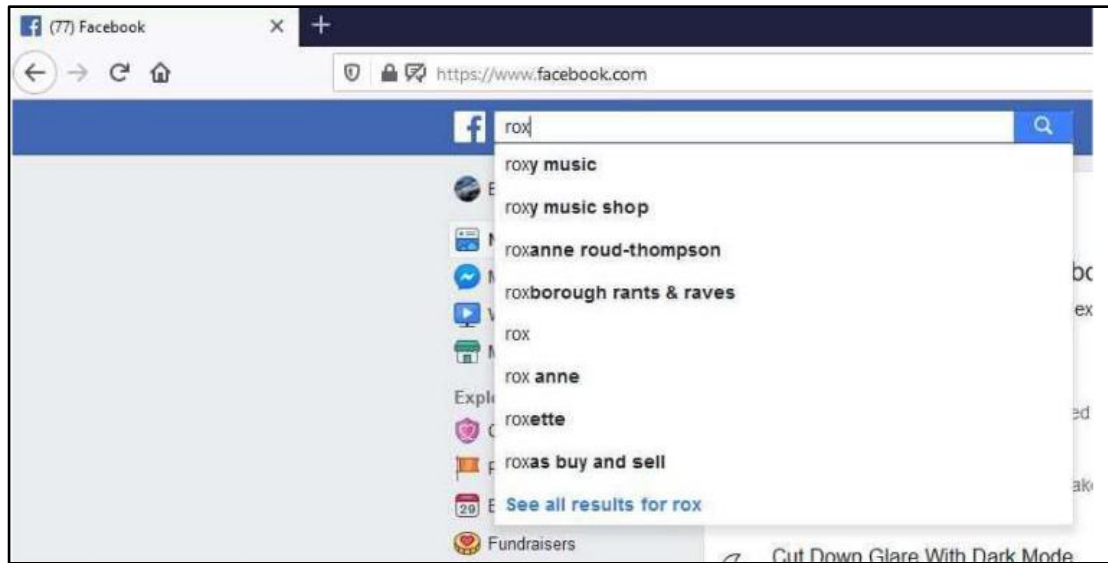
1[d] wherein the server system, while receiving said query messages, uses the input to query data available to the server system and send return messages to the client object containing results in response to the input; and

1[e] wherein, upon receiving a return message of the return messages from the server system, the client object tests the usability of the results in the return message by checking that the return message corresponds to the latest query, and if usability is established, the client object displays or returns at least some result data to the user.

Per the common specification, “the invention provides a session-based bi-directional multi-tier client-server asynchronous information database search and retrieval system for sending a character-by-character string of data to an intelligent server that can be configured to immediately analyze the lengthening string character-by-character and return to the client increasingly appropriate database information as the client sends the string” (’024 patent 8:31–38; *see also id.* at Abstract).

The common specification describes the invention as a “session-based” system. The Background section explains how most internet connections use the Hyper Text Transfer Protocol (HTTP), “which is inherently ‘session-less,’” in that “the server only checks the validity of the client or user input after the user sends back or submits an entire input form” (*id.* at 2:47–56). The invention addresses this issue through “sessions”: “In accordance with one embodiment of the invention the system is session-based, in that the system knows or recognizes when subsequent requests originate at the same Client” (*id.* at 12:11–13).

Before proceeding into further discussion of the claims, this order will get into the nuts and bolts of the accused instrumentality, Meta’s “Typeahead” system. Here is how it looks on the desktop version of Facebook:



(Peck Rep. ¶ 64, Homrig Decl. Exh. 3). A user here has typed “rox” into the Facebook search bar (an autocomplete query), and Typeahead has suggested several autocomplete query results, *i.e.*, suggestions for a complete query for potentially relevant pages, events, and applications, such as “roxy music,” “roxborough rants & raves,” etc. The autocomplete query results adjust as the user continues typing and revises her autocomplete query.

Typeahead is supported through both frontend functionality — provided by the user’s device — and backend functionality — provided by Meta’s servers. Typeahead first checks data on the frontend — stored in either a cache or a data structure called “bootstrap,” which contains a selection of pre-loaded potential search suggestions. If the frontend does not provide a sufficient number of suggestions, Typeahead goes to the backend, which Meta has dubbed “Unicorn” (so called because it provides functionality long sought after by Meta engineers). Unicorn provides functionality for many Facebook services, including Typeahead. *Important here, the parties’ experts agree that the entire string in the search bar is sent for each Typeahead request.* Once the search string reaches the backend, it is passed to Unicorn’s “top-level aggregator.” That delegates the Typeahead request to multiple “vertical aggregators” that search in parallel. Each vertical indexes a specific type of information, *e.g.*, people, events, groups, etc. So, the people “vertical aggregator” will search the people “index,” and a search string of “MA” might return the result “Mark Zuckerberg” from the people vertical. Also important for later, Unicorn uses a rote, call-and-response procedure for

answering requests sent from the frontend (Black Rep. ¶¶ 39–52, Homrig Decl. Exh. 8; Pei May 9, 2022 Dep. 7–8, Dkt. No. 193-4).

Having completed this high-level overview of the patents-in-suit and the accused system, this order will now turn to the specifics. This order follows full briefing and oral argument.

ANALYSIS

A patent infringement analysis involves two steps. The claim must be properly construed to determine its scope and meaning. Claim terms generally take “their ordinary and customary meaning,” that is “the meaning that the term would have to a person of ordinary skill in the art in question at the time of the invention.” *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312–13 (Fed. Cir. 2005) (en banc). Although construction begins with the claim language itself, “the specification is the single best” — and usually dispositive — “guide to the meaning of a disputed term.” *Network-1 Techs., Inc. v. Hewlett-Packard Co.*, 981 F.3d 1015, 1022 (Fed. Cir. 2020) (quoting *Phillips*, 415 F.3d at 1314–15).

The properly construed claim must then be compared to the accused device or process. To constitute infringement, an accused product must practice every limitation of a properly construed claim. *Tessera, Inc. v. Int’l Trade Comm’n*, 646 F.3d 1357, 1364 (Fed. Cir. 2011); *Carroll Touch, Inc. v. Electro Mech. Sys., Inc.*, 15 F.3d 1573, 1576 (Fed. Cir. 1993).

1. THE ACCUSED SYSTEM DOES NOT PRACTICE THE “QUERY MESSAGE” LIMITATION.

Meta argues that, in the patented system, the “query messages” sent between the client and the server contain “just the changes” in what the user has input into a growing search string, and that collateral estoppel bars MasterObjects’ arguments otherwise. MasterObjects opposes that sending “just the changes” constitutes merely one embodiment of the invention and that collateral estoppel does not apply. This order finds that collateral estoppel does apply and, even if it did not, that the claimed system’s “query messages” contain “just the changes.”

All of the claims-in-suit recite a limitation regarding the “query messages” sent from the client to the server as a user types out her autocomplete query:

- “whereby the query messages represent the lengthening string as additional characters are being input” (’024 patent, claims 1, 32, 35, 36, 37; ’628 patent, claim 13).
- “multiple query messages corresponding to multiple versions of said input” (’628 patent, claims 1, 25).
- “a request message containing a string representing an incomplete version of the search query” (’866 patent, claim 1).
- “sending a string representing an incomplete search query” (’073 patent, claim 1).

A practical example will prove helpful. A hypothetical Facebook user is looking for Mark Zuckerberg’s profile page. So, the user types in “M-A-R” into the search bar. The issue here is whether the client for the claimed system would send query messages of “M,” then “MA,” then “MAR” to the server; or, whether the system would send “just the changes” and the query messages would be “M,” then “A,” then “R” and so forth, with the characters then linked together on the backend by the server. Said another way, the dispute concerns whether the term is limited to a message comprising only the changes to an input string that were not sent in any previous consecutive query or may instead include the entire input string. The parties agree that Meta’s Typeahead system does not send just the changes. Instead, it sends the *entire* input in the search bar each time.

Judge Albright originally adopted MasterObjects’ preferred construction that query messages are not limited to sending just the changes. Of course, a claim construction adopted prior to transfer does not limit the transferee court’s prerogative in determining a different construction. *See In re Papst Licensing Digit. Camera Pat. Litig.*, 778 F.3d 1255, 1261 (Fed. Cir. 2015); *see also Vasquez v. Jan-Pro Franchising Int’l, Inc.*, 986 F.3d 1106, 1116 (9th Cir. 2021). Nor must Meta satisfy a motion-for-reconsideration standard. Indeed, this order would have nothing on which to base that evaluation as the minute order that adopted the preliminary constructions provided no analysis.

This order finds MasterObjects collaterally estopped from asserting its “query message” claim construction in light of the prior decision in *MasterObjects, Inc. v. Google, Inc.*, 2013

1 WL 2319087, at *5 (N.D. Cal. May 28, 2013) (Judge Phyllis J. Hamilton), *aff'd* 582 F. App'x
2 893 (Fed. Cir. 2014).

3 Collateral estoppel is not limited to patent claims that are identical. “Rather, it is the
4 identity of the *issues* that were litigated that determines whether collateral estoppel should
5 apply.” *Ohio Willow Wood Co. v. Alps South, LLC*, 735 F.3d 1333, 1342 (Fed. Cir. 2013)
6 (emphasis in original); *see also McQuillion v. Schwarzenegger*, 369 F.3d 1091, 1096 (9th Cir.
7 2004).

8 In *Google*, Judge Hamilton construed the “additional characters” term as found in the
9 ’529 patent and U.S. Patent No. 8,060,639, one iteration of which recited, for example:
10 “wherein each one of the plurality of queries are consecutive and together form an increasingly
11 focused query string for retrieving content from the server, and wherein each subsequent one
12 of the plurality of queries extends the query string in the user interface by one or more
13 additional characters.” *Google*, 2013 WL 2319087, at *11 (citing ’639 patent, claims 1, 13;
14 ’529 patent, claims 1, 44, 45). Judge Hamilton construed the term as follows: “only the
15 changes to the input string that were not sent in any previous consecutive query.” *Id.* at *12.
16 MasterObjects conceded noninfringement based on that construction, the district court entered
17 a stipulated final judgment, and the Court of Appeals for the Federal Circuit affirmed that
18 decision on appeal (Dkt. Nos. 54-19, 54-20). *See* 582 F. App'x 893 (Fed. Cir. 2014).

19 The ’529 and ’639 patents are in the same patent family as the patents-in-suit herein, and
20 they all share a common specification. The *Google* decision was based on the following
21 passage from that common specification (emphasis added):

22 If the results are not found in the cache, the Client Quester uses the
23 Client Controller to send the new input buffer to the Server
24 Quester, so that a new query can be executed (step **611**). To
25 support this, *the protocol of the present invention provides a
number of messages that allow the Client Quester to send just the
changes to the input buffer, instead of sending the entire input
buffer*

26 (’024 patent 20:11–17). The decision concluded “that the use of ‘the current invention’ here
27 indicates that the description is intended to apply to the invention as a whole, and not just a
28 single embodiment.” 2013 WL 2319087, at *12.

1 The claim-construction issue in this action is identical to the one addressed in *Google*:
2 whether the specification limits the claimed invention such that the client sends “just the
3 changes” to the server. As explained below, the claims-in-suit here do not differ substantively
4 from the claims considered in *Google* such that collateral estoppel would not apply. This order
5 finds MasterObjects collaterally estopped from asserting that the claims-in-suit do not send
6 “just the changes,” which leads to the conclusion that Meta does not infringe.

7 Assuming *arguendo* that collateral estoppel does not apply, this order reaches the same
8 conclusion as *Google*. “When a patent thus describes the features of the ‘present invention’ as
9 a whole, this description limits the scope of the invention.” *Verizon Servs. Corp. v. Vonage*
10 *Holdings Corp.*, 503 F.3d 1295, 1308 (Fed. Cir. 2007). The specification clearly states, “the
11 protocol of the present invention provides a number of messages that allow the Client Quester
12 to send just the changes to the input buffer, instead of sending the entire input buffer.”
13 MasterObjects asserts that the use of “allow” indicates that this description is permissive, not
14 mandatory. That misses the mark. The passage discloses the functionality of “the present
15 invention,” and then proceeds to indicate how that functionality *enables* the specific
16 embodiment being described. The description of the invention as a whole recites that the client
17 sends just the changes to the server.

18 The language of the claims-in-suit themselves do not alter this conclusion.
19 MasterObjects emphasizes the absence of the words “lengthens” and “modify,” which *Google*
20 cited as support for its construction. *See* 2013 WL 2319087, at * 12. The claims here use
21 different words, but they recite a system with the same functionality and do not indicate that
22 the system re-sends characters. The claims-in-suit use four iterations of the “query message”
23 limitation, which state that (emphasis added): the query messages “*represent* the lengthening
24 string”; the query messages “*correspond[]* to multiple versions of said input”; the “string
25 *represent[s]* an incomplete search query”; and that the “request message contain[s] a string
26 *representing* an incomplete version of the search query.” The use of words like
27 “corresponding” and “represent” indicate that the client sends information *about* the expanding
28 user search. Said another way, the claims state that a query message serves as an equivalent to,

1 or proxy for, what the user types into the search bar. We must read these descriptions in light
2 of the common specification. *Phillips*, 415 F.3d at 1315. The specification makes clear the
3 information communicated by the query messages about the user’s query are just the changes
4 not previously sent to the server.

5 The other intrinsic and extrinsic evidence supports this construction. *Nowhere* in the
6 patents-in-suit is it suggested that the client re-sends characters to the server. The Abstract
7 recites, in relevant part, that “[t]he invention provides a . . . system for sending a character-by-
8 character string of data to an intelligent server.” Character-by-character communications
9 inherently dictate sending only the changes. The specification states that “[t]he system’s
10 protocol is not restricted to sending single characters,” but the example thereafter only
11 contemplates “when a user replaces the contents of the entry with a new string” (’024 patent
12 12:5–10). Even in this example, the client only sends new characters (or a new set of
13 characters) to the server, never characters previously sent. These passages shed light on how
14 to read the term. *See Boss Control, Inc. v. Bombardier Inc.*, 410 F.3d 1372, 1378 (Fed. Cir.
15 2005).

16 All of these descriptions align because the invention sought to solve problems with auto-
17 complete functionality that was located on the client: “Today’s client-side auto-complete
18 functions are useful but very limited. The invention, however, vastly expands the usefulness
19 and capabilities of the auto-complete function by enabling the auto-complete data, logic and
20 intelligence to reside on the server, thus taking advantage of server-side power” (’024 patent at
21 6:40–45). Accordingly, “[t]he invention provides . . . a way to *synchronize* the data entered or
22 displayed on a client system with the data on a server system” (*id.* at 5:66–6:3, emphasis
23 added). The co-named inventor of the patents-in-suit (sans the ’073 patent), Stefan van den
24 Oord, testified: “[W]hat I would call the core of the invention, so things like non blocking,
25 asynchronous communication, sending *updates of queries and results*. That was all part of
26 Mark[Smit’s] original idea” (van den Oord Dep 57, Dkt. No. 54-5, emphasis added). As
27 Mr. van den Oord stated, the heart of the invention is sending *updates* of the queries, *i.e.*, only
28 the changes. *See Bradium Techs. LLC v. Iancu*, 923 F.3d 1032, 1044 (Fed. Cir. 2019).

1 The actual term “query message” is only used once in the specification and is of no help
2 to MasterObjects. The specification describes how the user can type an “a” and then a “b” into
3 the search bar, and proceeds to describe the system’s actions upon entry of the “b”: “As
4 before, a corresponding event arrives at the Server Quester. In this case, the Server Quester
5 may deduct [sic] that the input string represents a valid query and send the appropriate query
6 message ‘ab’ to the Service” (’024 patent 18:65–19:2). MasterObjects stresses that this
7 demonstrates that the claimed system need not send only the changes, it can send “ab” and not
8 just “b.” But Meta correctly points out that this passage describes a query message after it has
9 already reached the server, not during the pertinent transfer from client to server.

10 MasterObjects other arguments fail. The other district court decisions that have reviewed
11 MasterObjects’ patents did not consider this issue. *See MasterObjects, Inc. v. eBay, Inc.*, 2013
12 WL 1287428, at *7 (N.D. Cal. Mar. 28, 2013) (Judge Jacqueline Scott Corley); *MasterObjects,*
13 *Inc. v. Yahoo!, Inc.*, 2013 WL 6185475, at *5 (N.D. Cal. Nov. 26, 2013) (Judge Jeffrey S.
14 White). None of the patents asserted in *eBay* or *Yahoo* are asserted here. Neither decision
15 addressed the pertinent “just the changes” language from the specification, they merely held
16 that other terms in the claims at issue in those disputes should not be limited to a specific
17 embodiment. Nor did the Court of Appeals for the Federal Circuit affirm those decisions like
18 it did *Google*. MasterObjects’ citation to the PTAB decision initiating *inter-partes* review for
19 the ’024 patent carries little persuasive weight; it was not a final decision, and it employed the
20 broadest reasonable interpretation standard (Dkt. No. 55-10 at 8). MasterObjects’ statements
21 in that IPR proceeding that query messages are not limited to sending only the changes (Dkt.
22 No. 56-3) — statements made years after *Google* — did not foster intrinsic evidence sufficient
23 to affect the outcome here.

24 Because the parties do not dispute that the accused system does not send “just the
25 changes,” this order concludes that Meta does not infringe the claims-in-suit.

2. **THE ACCUSED SYSTEM DOES NOT PRACTICE “ASYNCHRONOUS” CLIENT-SERVER COMMUNICATION.**

Meta also asserts a noninfringement theory pursuant to Judge Albright’s construction of the term “asynchronous.” MasterObjects argues that Meta improperly construes “asynchronous” to require the server to be able to *initiate* communications with the client. This order finds Meta’s argument on this point persuasive.

All the claims-in-suit use the term “asynchronously” or “asynchronous” to refer to the communication between the server and the client. Asynchronous in the general, computer programming sense broadly refers to “something that is not depending on timing” (Dkt. No. 55-16). Here are the parties’ proposed constructions and the construction adopted in Texas for the term as used in the claims-in-suit:

MasterObjects	Meta	W.D. Tex. Construction
Plain and ordinary meaning. Alternatively: “each side of the communication is free to communicate without waiting for the other side” Alternatively: “each side of the communication can communicate with the other side in a non-blocking manner”	“Both the client and the server can initiate communications at any moment in time”	“Each side of the communication is free to communicate without waiting for the other side”

(*see* Dkt. Nos. 62, 69; Homrig Decl. Exh. 2). Like the “query message” term, this order is not bound to the construction adopted in Texas. For the reasons that follow, however, this order need not adjust the prior construction because it adequately captures the nature of the term, so we look to the specific issue raised by Meta.

Meta asserts the accused system cannot initiate a communication from the server to the client. Accordingly, it contends that its system does not infringe because all the claims-in-suit require asynchronous communications between the server and client *and further contends that asynchronous means that the server must be able to initiate communications*. This order agrees.

As recited in the specification, “[t]he system is bi-directional and asynchronous, *in that* both the Client and the Server can *initiate communications at any moment in time*” (’024 patent 12:24–26, emphasis added). The common specification expressly states that a server communicating asynchronously is able to initiate communication with the client. The accused system cannot initiate communications. To this, MasterObjects makes the following arguments.

Primarily, MasterObjects contends that the passage above describes a non-limiting embodiment. The specification reminds the reader throughout that it describes a preferred embodiment, “QuestObjects” (*e.g.*, *id.* at 9:53–54; 31:38–41). To that end, MasterObjects argues that the capitalized “Client” and “Server” in the passage Meta cites are embodiment specific. The specification, however, does not define *capital* “Client” and “Server,” but lower-case iterations of the words: “The terms “client” and “server” are used herein to reflect a specific embodiment of the invention” (*id.* at 11:50–57). Because the specification only defined lower-case “client” and “server,” this order finds that this passage of the specification does not expressly refer to a particular embodiment. Moreover, even accepting MasterObjects’ position, the passage describes a specific embodiment with bi-directional and asynchronous functionality — it does *not* indicate that the term asynchronous itself has alternative, embodiment-specific attributes. Rather, the specification off-sets the term and then provides additional, term-specific description independent of any embodiment. *See TriStrata, Inc. v. Microsoft Corp.*, 594 F. App’x 653, 655–56 (Fed. Cir. 2014); *Honeywell Int’l, Inc. v. ITT Indus., Inc.*, 452 F.3d 1312, 1318–19 (Fed. Cir. 2006).

In fact, MasterObjects also used the same description when distinguishing the “Purcell” prior art while prosecuting the grandfather ’529 patent. It explained that “Purcell does not disclose a *session-based* environment, wherein a communication protocol provides an *asynchronous session-based connection* between the client system and the server system” (Dkt. No. 54-21 at 37, emphasis in original). MasterObjects described asynchronous the same way it does in the specification: “[S]ince the system is *asynchronous*, both the client and the server can initiate communications at any moment in time” (*id.* at 36, emphasis in original).

1 MasterObjects has remained consistent with how it uses the term. *See Teva Pharms. USA, Inc.*
2 *v. Sandoz, Inc.*, 789 F.3d 1335, 1343 (Fed. Cir. 2015).

3 MasterObjects argues that this description of asynchronous has no bearing because
4 “[e]very asserted claim says the client sends messages, and the server responds to the client’s
5 messages” (MO Claim Const. Resp. Br. 22). Claim terms, however, must always be read in
6 light of the specification, and should not be rendered superfluous by a particular construction.
7 *Merck & Co., Inc. v. Teva Pharms. USA, Inc.*, 395 F.3d 1364, 1372 (Fed. Cir. 2005); *Phillips*,
8 415 F.3d at 1315. The claimed server does not merely receive and respond to the query
9 messages, it “*asynchronously* receive[s] and respond[s] to the query messages.” To give
10 meaning to the term in this context, we turn to the common specification. It teaches: “An
11 example of a communication initiated by the Server is updating the information provided to the
12 client. Because the system is session-based it can keep track of database information that has
13 been sent to the Client. As information changes in the database, the Server sends an updated
14 version of that information to the Client” (’024 patent 12:33–38; *see also id.* at 17:54–57).

15 This passage explains that the claimed server will, *without prompting*, initiate a further
16 response to the client’s request and provide additional suggestions. The specification
17 demonstrates that a proper construction should not recast the term “asynchronous” merely
18 because the words “receiving” and “responding” are used in a claim. Indeed, the specification
19 further explains that “[d]ata can also be presented to a client without user input, i.e. the data are
20 automatically pushed to the client” (*id.* at 6:15–16; *see also id.* at 8:55–57 (same)). The
21 language of the claims does not indicate that only the client initiates communications while the
22 server lacks such functionality. Such a construction would impermissibly divorce the claim
23 language from the specification and prosecution history or would read the term out of the
24 claims entirely. *See UltimatePointer, LLC v. Nintendo Co., Ltd.*, 816 F.3d 816, 823–24 (Fed.
25 Cir. 2016); *Decisioning.com, Inc. v. Federated Dep’t Stores, Inc.*, 527 F.3d 1300, 1308 (Fed.
26 Cir. 2008) (per curiam).

27 Three prior district court decisions have construed “asynchronous” for other
28 MasterObjects’ patents not asserted here. *See eBay, Inc.*, 2013 WL 1287428, at *7; *Google*,

2013 WL 2319087, at *5; *Yahoo*, 2013 WL 6185475, at *5. While *Google* construed the term to mean “a connection that allows both the client and the server to initiate communications at any moment in time within a session,” the other two decisions reasoned that the server’s ability to initiate communications was one embodiment of the invention, and that the claims-in-suit therein did not recite that requirement. Despite this, MasterObjects has asserted that *Google*’s construction is consistent with its position in this action (MO Claim Const. Br. 16). Given the record here, this order finds *Google* the most persuasive of the three.

MasterObjects has used shifting-sands tactics regarding session-based systems to avoid the conclusion that the claimed server can initiate communications. At claim construction in this action MasterObjects asserted:

Facebook’s argument that the intrinsic evidence defines “asynchronous” to require server-initiated communication is based on a faulty premise, that ‘the asserted patents are directed to a system in which the client and server form a special connection — a “session.” That is not correct. None of the asserted claims recite a “session,” and the specification makes clear that the claims are not limited to the session-based embodiments

(MO Claim Const. Reply Br. 13). It made this argument to distinguish the intrinsic evidence discussed herein as specific to a session-based embodiment (MO Claim Constr. Resp. Br. 25). Now, at summary judgment, MasterObjects takes a contrary position: “The claims themselves **require** that the client initiate the conversation (*the user search session*)” (Opp. 11, bold in original, italics added). By making this assertion now, MasterObjects undermines its claim-construction arguments.

Ultimately, whether the claims-in-suit recite an implicit “session” limitation does not change the outcome here. As MasterObjects has acknowledged, “a communication is ‘asynchronous’ if *both sides* are free to talk without relying on a clock or other coordination mechanism to synchronize their communications with one another” (MO Claim Const. Br. 26, emphasis added). Because the claims-in-suit require the server to be “free to communicate” with the client, it necessarily follows that the server must itself be able to initiate communication. MasterObjects has provided no evidence of that functionality in the accused system.

1 The evidence supports the conclusion that Typeahead is a rote call-and-response system
 2 where the server is not free to communicate spontaneously. MasterObjects supports its
 3 infringement theory with the report of its expert John Peck, where he opined that the
 4 Typeahead “*client object* can send evolving queries without waiting for a response to prior
 5 queries” (Opp. 12, emphasis added). But that addresses only half the problem — even
 6 MasterObjects recognizes the claims-in-suit mandate that *both* sides must be “free to talk.”
 7 Upon review, Expert Peck provided no evidence demonstrating that Typeahead’s *server* can
 8 communicate without waiting for a corresponding communication from the client (*see* Peck
 9 Rep. ¶¶ 56–90). The report walks through how Typeahead generates autocomplete queries and
 10 results as the user types in “Roxy music” character-by-character into the search bar. For each
 11 successive character, the client sends a new message to the Typeahead backend using the
 12 HTTP GET method. Expert Peck then simply tendered the conclusory statement “[t]he server
 13 system receives and response [sic] to query messages asynchronously” (*id.* at ¶ 65). He
 14 offered no insight into *how* the server asynchronously communicates with the client once it
 15 receives each request. The example Expert Peck used shows the accused server providing rote
 16 responses to each HTTP Get request, not a server that can freely and spontaneously
 17 communicate without waiting for the client, as Meta’s noninfringement expert John Black
 18 explained in his rebuttal report (Black Rep. ¶¶ 57–64, Homrig Decl. Exh. 8).

19 Expert Peck concluded his analysis with a passing reference that the client sends an
 20 “AJAX” request to the Typeahead backend. He explained that “AJAX is a technique for
 21 sending and receiving data asynchronously,” citing the Wikipedia entry for AJAX (which
 22 stands for “Asynchronous JavaScript and XML”) (Peck Rep. ¶ 90). But as Expert Black
 23 responds, “in the context of AJAX, asynchronous does not refer to the client and server being
 24 ‘free to communicate without waiting for the other side,’ but rather a different meaning of
 25 asynchronous, whereby the *client* can send and retrieve data in the background without
 26 interfering with the display and behavior of the user interface” (Black Rep. ¶ 61, emphasis
 27 added). Expert Peck’s reference to AJAX does not reflect the use of asynchronous in the
 28 claims-in-suit. It does not shed any light on how the server communicates asynchronously.

1 Upon review, this order finds that MasterObjects has not raised a genuine dispute of material
2 fact regarding Typeahead's use of AJAX.

3 MasterObjects similarly cites testimony from the transcript of Meta's Rule 30(b)(6)
4 witness Markus Messner-Chaney, who testified that the accused instrumentality uses AJAX
5 and that "[t]he client can respond to other inputs, and if that happens, it will check bootstrap
6 [on the frontend]. It will send a request. At some point, there may be that response comes in .
7 . . ." (Messner-Chaney Dep. 77–87, Hosie Decl. Exh. D). But all Mr. Messner-Chaney stated
8 is that the client will respond as the user continues typing her query. He provided no testimony
9 that the Typeahead server can freely communicate without waiting for the client.

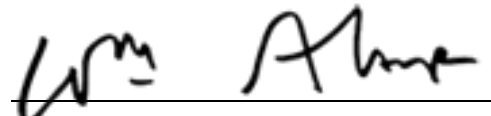
10 In sum, this order finds that the accused system does not infringe because it does not
11 practice the "asynchronous" claim term.

12 CONCLUSION

13 For the foregoing reasons, Meta's motion for summary judgment of noninfringement is
14 **GRANTED**. The Court would like to thank both Attorney Tiffany Weston and Attorney Darrell
15 Atkinson for their excellent arguments at the hearing.

16 **IT IS SO ORDERED.**

17
18 Dated: October 20, 2022.

19
20 
21 WILLIAM ALSUP
22 UNITED STATES DISTRICT JUDGE
23
24
25
26
27
28